DOMOS

A nonlinear transient simulation and
optimization program for
integrated MOS circuits

Version 7/5
(03.12.1982)
User's manual

H. Sibbert

Universitaet Dortmund
Lehrstuhl Bauelemente der Elektrotechnik

General description

DOMOS is a transient simulation program for integrated MOS circuits. DOMOS uses trapeziodal integration, and the system of linearized equations is solved by means of an Gauss-Seidel iteration. Therefore DOMOS is a fast and accurate simulator with low storage requirements.

An additional feature of DOMOS is the implemented optimizing processor which is very efficient. It performes the optimization of many circuit parameters in a wide range of values even more than one at the same time. For optimization the algorithm of Fletcher - Powell is used, which is detailly described in [2].

For further information about DOMOS please refer to [1].

Input language

The input to DOMOS is free of format. Only coloums 1 to 72 are read. The only delimiter is the blank (one or more).

The input deck is devided into a number of blocks. The sequence of these blocks is arbitrary with the exception of the lines EXECUTE, RESET, and END. Some blocks consist of one line only. Blocks consisting of more than one line have to start with their block header. DOMOS accepts the following block headers :

Block Header	Single	Requested
TITLE	**	
NOSUMMARY	**	
TEMPERATURE	**	

CIRCUIT		**
INITIAL		
MILS	**	
MICROAMPS	**	
PARAMETERS		
FINAL VOLTAGES	**	
CHECK		
ERROR	**	
ITERATIONS	**	
TIMER	**	**
WIDTH	**	
OUTPUT		**
OPTIMIZE		
UNLIMITED	**	
EXECUTE	**	**
CONTINUE	**	
RESET	**	
END	**	**

now a description in detail of all control cards is following. if the card is marked as single the block consists only of the block header. the control statements are enclosed by colons. the colons are not part of the control statement they are only used for a better readability. [....] fields are optional, (....) fields are requested.

1.0.0. "title [text]"

the field text is printed as a headerline onto each page of simulation output. a number up to three "title" cards is accepted.

2.0.0 "iterations (nr)"

nr is the permitted number of iterations in the gauss - seidel algorithm. nr has to be a positive integer number.
default value : 20

3.0.0. "nosummary"

usually at beginning of domos execution the input deck is printed. this control card suppresses the printing and also that of the simulation summary.
useful for paper economizing!

4.0.0. "temperature (temp)"

temp is the temperature in deg celsius at which the simulation will be performed. is the card omitted the default value for temp is taken : 25 deg celsius.
remember :the parameter values have to be specified with their 25 deg values. see (5.0.0. parameter card)

5.0.0. "parameters"

this card opens the block for the input of the semiconductor - device model parameters. the cards following the block header "parameters" have the format :

"lab (parm1) (val1) [parm2] [val2] [parm3] [val3] ...."

lab : label of the parameter set (this label is referenced on the card which specifies the concerning semiconductor device).
parm : name of the parameter.

val : value of the parameter at 25 deg celsius.

5.1.0. table of parameter names

name	comments
chanl	channel type (p or n)
body	body contact of transistors
nb	substrat doping
tox	gateoxide thickness
vt0	threshold voltage , >0 enhancement - <0 depletion transistor
tcvt	temperature coefficient of vt0 [v/deg ^cc]
cgox	gate capacity
b0	zero field mobility
theta	mobility reduction coefficient
ec	critical field
lov	gate-source/drain overlapping
k1	substrate effect constant
cj	junction capacity at 0v
cexp	exponent for junction capacity formula.
fw	channel width correction
fl	channel length correction
phis	surface inversion potential
f	correction factor for k2
k2	channel length modulation factor
eg	characteristic field
a	length correction (unused)
b	width correction (unused)
is	diode saturation current
dbreak	didoe breakthrou voltage
dres	diode leakage resistance

not initialized parameters are set to default values. for a table of parameter defaults see 20.0.0.

6.0.0. "initial"

this controlcard opens the block where the node voltages are initialized if this is wanted. the following cards have the format :

"n(nr) (u) <u> <u>....."

nr : number of the node which will be initialized with the first voltage value on the card.if more than one value is specified the nodes are initialized in ascending order beginning at node nr.

u : voltage value

nodes that are not explicitly initialized in this block have the voltage 0.0 v at the start of the simulation.

7.0.0. "final voltages"

by inserting this controlcard all node voltages are printed with their values at the end of the simulation. so simulation can be continued with this voltages as initial ones. (see "initial" card 6.0.0. , "continue" card 16.0.0.)

8.0.0. "circuit"

this card opens the block where the circuit description is processed. the elements of the circuit are defined here. no special order of the following cards is requested.

8.1.0. resistors

```
"r(nr) (c1) (c2) (val)"
```

```
r7 n1 e3 10
```

nr : number of ohmic resistor (1..50)
c1,c2 : connected from c1 to c2
val : resistance

8.2.0. capacitors

```
"c(nr) (c1) (c2) (cf) [(cv0) (lab)]"
```

```
c21 n4 n0 100
c82 e2 n0 2 2.2 p
```

nr : number of the capacitor (1..100)
c1 : anode terminal
c2 : cathode terminal
cf : capacitor value (fixed capacitor)
cv0 : junction capapacity at 0 v barrier layer voltage
lab : label of the parameter set that contains the diode parameters belonging to the junction capacity.

8.3.0. diodes

```
"d(nr) (lab) (c1) (c2) (area)"
```

```
d11 p e1 n100 200.5
```

nr : number of the diode (1..50)
lab : parameter set lab is used in the diode model
c1 : anode terminal

1

c2 : cathode terminal
area : cross section of the diode

8.4.0. transistors

```
"t(nr) (lab) (c1) (c2) (c3) [c4] (w) (l) [sd] [dd]"
```

```
t129 l e1 r12 c3 200 200
```

nr : number of the transistor (1..200)
lab : parameter set lab is used in the transistor model
c1 : gate
c2 : source
c3 : drain
c4 : body (if omitted "body" - parameter is necessary , see 5.1.0.)
w : channel width
l : channel length
sd : source diffusion region area (default : 0.0)
dd : drain diffusion region area (default : 0.0)

8.5.0. current sources

```
"j(nr) (c1) (val)"
```

nr : number of the current source (1..20)
c1 : node into which the current flows
val : amperage

8.6.0. voltage sources

```
"e(nr) [typ] (a) [(td) [(vo) [(tw) [(tr) [(tf) [(tp)] [op] [src]]
```

```
e1 5
e2 linear 4 1 100 200 50 50 500
e7 expo 5 0 0 200 50 50 500
```

a	:	amplitude		
td	:	delay time		
vo	:	offset voltage		
typ	:	linear	expo	sine
tw	:	impulswidtdh	impulswidth	time shift
tr	:	rise time	tcr=tr/2.2	linear damping coefficient
tf	:	fall time	tcf=tf/2.2	exponential damping coefficient
tp	:	period	period	period

op : superposition operator + - *

src : superposition source, any previously defined voltage source

default values : tw,tp : 10**36 , tr,tf : 0.0

1

remarks

linear : the universal linear waveform is trapezoidal.

time	:	0	td	td+tr	td+tr-tf	td+tw	td+tp
voltage	:	vo	vo	a+vo	a+vo	v0	vo

if tp [tw the following impuls starts in spite the preceeding on has not finished.

sine : the amplitude is time dependent in the following way :

$$a(t) = (a + tr\ (t - td))\ exp\ (tf\ (t - td))$$

remember : tr , tf have to be negative numbers to get decreasing amplitude.
tw : time shift is phase shift devided by 2 * pi * frequency. at t = td + tw a period starts with voltage value vo.

expo : the linear rise and fall of the impulse is replaced by an exponential one with the time constants tcr (rise) , tcf (fall).

8.6.1. superposition of voltage sources

to realize complex voltage functions sources may be superpositioned, by adding the superpositon operator and source to the definition of the voltage source to be superpositioned. the source, which might be added or multiplied must have a lower number and must be already defined.

```
example
e1 linear 1 0 0 10 1 1 100
e2 expo 5 0 0 200 50 50      + e1
e3 sine 5 0 2 20 0 0 500     * e2
```

9.0.0. "timer (tstart) (tstop) [delmin] [delmax]

tstart : starting time of simulation (.ge.0)
tstop : stop time of simulation
delmin : minimal stepsize (default : (tend-tstart)/10**5)

1

delmax : maximal stepsize (default : delmin*10**3)

this card defines the simulation interval. absence would cause a fatal error and program termination.

10.0.0. "mils"

switches the length unit to mil (milli-inch , 25.4 um) parameter default values are converted by the program corresponding to this selection.

11.0.0. "error (typ) (err)"

typ : "absolute" or "relative"
err : error

if "absolute" is selected "err" is the absolute voltage error , if "relative" is taken "err" is the relative voltage error in units of 100 % (.001 means 0.1 % error). is one node voltage less than 1 v "absolute" is taken anyway.

12.0.0. "check"

this card opens the block where two check facilities are handled. the following cards have the format :

"maxmin (var1) [var2] [var3] ......"

the maximium and minimum values and the time of their occurance during the simulation are printed for the variables var1,var2,... for a list of valid output variables see output card (13.3.0.).

"elements (el1) [el2] [el3] ....." or
"elements all"

this card causes the interconnections of the element group el1,el2,... to be printed in the simulation summary. if "all" is specified this is done for all element groups that can be simulated with domos.

el	group	el	group
r	resistors	c	capacitors
t	transistors	n	nodes
d	diode	e	voltage source
j	current sources		

1

13.0.0. "output"

this card opens the block where the output processor gets the information about the requested output mode and variables. the absence of this card in the input deck will cause a fatal error and program termination.

13.1.0. fix output

"fix [plot] [print] [120] [step]" or
"fix cancel"

plot : output is a printerplot
print : output is a printed table

120 : limits the plot width to 120 characters per line

at least one item "plot" or "print" must be specified.

if the fix output mode is taken the values of the requested variables are printed / plotted at every time t = n * step , n = 0,1,2,... the following card(s) contain the list of output variables :

"v(1) [[(v2) (v3) .... (v10)] (min) max)]"

v1 .. v10 are the variables that are printed / plotted by the output processor. up to 10 variables per card are permitted. 10 cards can be processed but the total number of output variables must not exceed 40.

"min","max" are optional and define the plot bounds. if they are not specified the program computes the plot bounds to fit the data points into the plotting area.

all variables listed on the same card will be printed / plotted in one printer table / plot. so ensure that they are in the same order of magnitude if a plot is wanted.

the "cancel" mode cancels the fix - output in a modification run (see 15.1.0.)

the second output mode that can be selected is the vari mode :

13.2.0. vari output

"vari [print] [plot] [120]"

the difference to the fix mode is that the variables are printed or plotted at every internal timestep where simulation is performed. the vari mode should not be used because no significant time scale is generated. the vari mode produces large amount of output. vari mode and "opimize" conflict and must be not used in one input deck (see 14.0.0.)

1

13.3.0. valid output variable names

r1...r50	current through resistor 1..50
t1...t200	current through transistor 1..200
d1...d50	current through diode 1..50
c1...c200	capacity of node 1..200
n0...n200	node voltage of node 1..200
e1...e20	voltage of source 1..20
j1...j20	current of current source 1..20
w1...w200	power dissipation of transistor 1..200
w0	power dissipation of the total circuit

14.0.0. "optimize"

this card forces the program to perform circuit optimization. therefore an efficient algorithm of fletcher and powell is implemented. it performs the minimum searching of a function in several variables. up to 20 independent variables can be optimized at the same time. also so called dummy variables can be among this set. the error function which should be minimized can be devided into several partial objective functions (up to 20).

if several variables of the circuit should be optimized at the same time you insert the concerning cards described below. the program will optimize them to obtain a minimal error function by increasing or decreasing the value of one variable independently from all other ones. this is done for all variables to be optimized. if a defined relation between two (or more) variables has to be maintained this can be done by the use of a dummy variable.

14.1.0. objective functions (define error function)

```
    "obj [typ] (v1) [v2] [w0] [t1 w1] [t2 w2]....[t9 w9]"
obj - d1 t3 0 200 1
obj n4
```

typ : can be "+" or "-"
v1,v2 : output variable name (see 13.3.0.) , default v2 : n0
t1..t9 : time bounds (default : 10**60)
w0..w9 : weight factor (default : 1)

this card defines the following function :

```
          obj:= time integral t=0...t1 : (v1-v2)**2*w0
              + time integral t=t1..t2 : (v1-v2)**2*w1
              + ......................................
              + ......................................
              + time integral t=t8..t9 : (v1-v1)**2*w8
              + time integral t=t9..ts : (v1-v2)**2*w9
```

ts....end of simulation
if typ "+" is selected only positive differences (v1-v2) are integrated (in analogy typ "-"). the following cards request the optimization of the circuit elements.

14.2.0. resistors , diode cross section , current sources

```
    "(typ)(nr) (min) (max)"

r10 10 20 d14 1 1.5
```

typ : r...resistor
      d...diode
      j...current source

nr : number of the element
min : minimum
max : maximum of the element parameter

14.3.0. transistors , capacitors

since each transistor and capacitor have two element parameters which can be optimized a different statement format is used.

```
"(typ)(nr) (name) (min) (max) [(name) (min) (max)]
```

```
t159 w 6.25 15 l 7.5 20
t159 l 7.5 20 w 6.25 15
t187 l 20 24
c33 f 1.1 v .3 .9
```

typ : t...transistor
c...capacitor
name : w/l...width/length of a transistor
f/v...fixed or variable part of a capacitor
min : see 14.2.0.
max : see 14.2.0.

14.4.0. voltage sources

```
"e(nr) [srnr] (name) (min) (max) [(name) (min) (max)].....
```

nr : number of the voltage source

srnr : number of the partial source if superposition was used (see 8.6.1.) default : 1
min : see 14.2.0.
max : see 14.2.0.
name : a...amplitude
d...delay time
o...offset voltage
w...impulse width (time shift)
r...rise time (linear damping)
f...fall time (exponential damping)
p...period

14.5.0. semiconductor devices

```
"parm (lab) (name) (min) (max) [(name) (min) (max)]....."
```

```
parm n vt0 .5 1.5 tox .07 .12
```

lab : label of the parameter set
name : see 5.1.0.

14.6.0. dummy variables

```
"p(nr) (min) (max) (init)"
```

```
p3 1.2 1.5 1.22
```

nr : number of the dummy variable
min : see 14.2.0.
max : see 14.2.0.

init : initial value of the dummy variable (1..10)

you declare an element parameter (circuit parameter) as a function of a dummy variable by replacing the (min) (max) fields by the following declaration sequence :

"p(nr) [(a0) [(a1) [(a2) [(a3)]]]]"

please note that the above sequence is no control card for its own. it can only appear as a part of control statements described in 14.2.0. , 14.3.0. , 14.4.0.

examples

```
r5 p3
c7 p3 1 2
e19 4 a p4 0 0 0 1.4 d 2 4
```

the actual value of the circuit parameter (element) x is

$$x = a0 + a1 * p + a2 * p ** 2 + a3 * p ** 0.5 \quad (1)$$

default : a0 = a2 = a3 = 0 , a1 = 1 .

so all relations which can be described with equation (1) can be maintained between the optimized variables. up to 100 circuit parameters (elements) can be declared as dummy variable functions.

14.7.0. optimize control cards

"optcntl (parm) (val) [(parm) (val)]"

parm : limit : val...number of search directions (default 4)
step : val...stepsize in search algorithm

15.0.0. "execute"

this card terminates the input sequence for a simulation and starts execution.

15.1.0. modification run

domos offers the possibility to change or to add circuit elements without repeating the whole circuit description. therefore the changed or added cards are inserted after the "execute" statement of the main simulation followed by a second "execute" card. block headers are required if the new cards are members of a block (e. g. circuit description after "circuit" - header) .

limitations

1) deletion of elements is only possible by redefining them and connect all terminals to ground.

2) no additional voltage sources are possible.
3) the number of superpositioned voltage sources may not be changed.
4) if the body terminal is changed in the parameter list it is only supplied to transisistors which are redefined or added in this simulation.
5) parameter cj=0 clears sd and dd transistor specifications (see 8.4.0.) .

changing of the output variable list is done by entering the whole new list.

16.0.0. "continue"

this card forces the program to start the simulation with the node voltages obtained with the preceeding one (in the same computer run) .

1

17.0.0. "reset"

the simulation of a completely new circuit in the same run is possible after inserting this card.

18.0.0. "end"

terminates the input stream.

19.0.0. comment cards

a "$" or "!" character anywhere on the card indicates that the following text is a comment. lank cards cause blank lines in the printout.

1

20.0.0. parameter default values

in the following table the parameter default values are listed. mathematical terms are assumed in fortran formula notation.

NAME	VALUE
CHANL	P
BODY	NO SUBSTRAT CONTACT
NB	5E14
TOX	.12
VTO	1.0
TCVT	0.0
CGOX	3.3E-5 / TOX
BO	20 * CGOX P-TYPE 50 * CGOX N-TYPE
THETA	4E-3 / TOX
EC	80 * CGOX / BO
LOV	2.5
PHID	0.7
K1	SQRT(3.403E-23 * NB) / CGOX
CJ	K1 * CGOX / (2 * SQRT(PHID))
CEXP	0.5
FL	1.5
FW	0.0
PHIS	2 * PHIF
F	2.0
K2	2.124E-4 / (F * K1 * CGOX)
EG	2 * SQRT(PHID) / (K2 * F)
A	.0
B	0.0
IS	1.0E-9
DBREAK	25.0
DRES	1.0E20

```
PHIF = 8.62E-5 * T * ALOG(NB / NI)                 FERMILEVEL
NI = 6.8125E15 * T ** (3/2) * EXP(-6438.5/T)       INTRINSIC
                                                   CARRIER
```

1

21.0.0 VAX/VMS implementation notes

Domos needs a single parameter for execution, that is the name of the input data-set. The user is prompted for the file name or this specifier can be include in the commandline if DOMOS is defined as a foreign command. The default extension used with this file is ".DOM". Optionally a qualifier "/LIST" can be given anywhere in the commandline, which cause the printed output to be directed into a file instead of using the SYS$OUTPUT device. (This file has the same name as the inputfile, but uses a ".LIS" extension. Any output requested within the "OUTPUT"-section of the domos input-set will be stored in files with an ".HP2" extension, for further processing. The first file will have the same name as the input-file, while others (e.g. produced by modification-runs with "KEEP"-option) will have a two digit hexadecimal number added to the original name, if the original name is longer than 7-characters it will be truncated, to avoid illegal filenames on the VAX-system.

The Program is delivered as a single fortran-source file. to install it, it has only to be compiled and linked.